

Hoja de Trabajo: Rescate Ágil del Proyecto POS

Nombre del Alumno: Anthony Chaguanac

Fecha: 18/01/2026

Grupo: _____

Instrucciones para el Alumno

Paso 1: Identifica los problemas

Antes de responder las preguntas, revisa el contexto del proyecto POS desarrollado con J2EE/Java EE (Servlets, JSP, EJB, JDBC, PrimeFaces, JPA, y despliegue en WebSphere Application Server). Escribe al menos 3 problemas relacionados con el desarrollo en cascada que afectaron el proyecto. Relaciona estos problemas con los principios ágiles que podrían haberlos evitado. Si no sabe qué es el modelo waterfall (hacer una investigación rápida)

Problemas identificados del desarrollo en cascada o waterfall (3 puntos):

1. Cliente retrasó la aplicación POS hasta el final del proyecto, provocando una interfaz confusa.
2. Los requisitos rígidos definidos al inicio, impidieron agregar funcionalidades clave.
3. La documentación extensa retrasó el desarrollo y no reflejó el valor real del software funcional.

Objetivo

Analizar un proyecto de punto de venta (POS) desarrollado en cascada/waterfall, identificando sus problemas y proponiendo cómo los doce principios del Manifiesto Ágil habrían evitado estas fallas. Responde en los espacios proporcionados, conectando cada problema con un principio ágil y explicando cómo lo aplicarías, considerando el contexto tecnológico de J2EE/Java EE (Servlets, JSP, EJB, JDBC, PrimeFaces, JPA, y despliegue en WebSphere Application Server).

Contexto

Escenario del Proyecto: Un equipo de 6 desarrolladores creó una aplicación de punto de venta (POS) para una cadena de tiendas minoristas, usando Java con JPA para la lógica de negocio, PrimeFaces para las pantallas (basadas en JSP y Servlets), y desplegada en WebSphere Application Server (WAS), un servidor con licencia comercial. El proyecto siguió el modelo en cascada durante 9 meses:

- Requisitos (2 meses): Documento de 60 páginas detallando gestión de inventario, ventas y reportes, aprobado por la tienda.
- Diseño (2 meses): Arquitectura (entidades JPA, servicios EJB, pantallas PrimeFaces) diseñada sin consultar al cliente.
- Desarrollo (3 meses): Lógica con JPA y pantallas con PrimeFaces implementadas sin mostrar avances.
- Pruebas (1.5 meses): Pruebas revelaron necesidad de interfaz PrimeFaces más simple y función de descuentos rápidos no incluida.
- Entrega (0.5 meses): App desplegada en WAS, pero el cliente rechazó el entregable por interfaz confusa, reportes inadecuados, y desconocimiento de costos de licencias comerciales de WAS, negándose a pagar. Se necesitaron 2 meses extra para ajustes.

Problemas Identificados y Principios Ágiles

1. El cliente no vio la app hasta el final, causando una interfaz desalineada.

2. Los requisitos rígidos impidieron agregar funciones como descuentos rápidos en JPA.
3. La documentación extensa retrasó el desarrollo con Java y PrimeFaces.
4. La entrega tardía del proyecto y los costos ocultos de licencias del servidor de aplicaciones llevaron al rechazo del cliente y la no facturación del proyecto

Principios Ágiles

1. Satisfacer al cliente mediante la entrega temprana y continua de software valioso.
2. Aceptar cambios en los requisitos, incluso en etapas tardías.
3. Entregar software funcional con frecuencia (cada 2-4 semanas).
4. Colaboración diaria entre interesados del negocio y desarrolladores.
5. Construir proyectos con individuos motivados, dándoles confianza y apoyo.
6. Utilizar la comunicación cara a cara siempre que sea posible.
7. El software funcional es la principal medida de progreso.
8. Promover un desarrollo sostenible con un ritmo constante.
9. Buscar la excelencia técnica y un buen diseño.
10. Maximizar la simplicidad enfocándose en el trabajo esencial.
11. Permitir que los equipos autoorganizados creen las mejores arquitecturas.
12. Reflexionar regularmente para mejorar la efectividad del equipo.

Preguntas para Analizar

Responde en los espacios proporcionados, conectando cada problema con un principio ágil y explicando cómo lo aplicarías al proyecto POS (Java, JPA, PrimeFaces, WAS, costos de licencias). Usa al menos cinco principios diferentes en total. Escribe respuestas claras y concisas.

1. ¿Cómo habría evitado un principio ágil que el cliente no viera la app hasta el final, asegurando una interfaz alineada? (1 Punto)

Principio Ágil: Satisfacer al cliente

Aplicación:

Se habrían implementado entregas de incrementos funcionales cada 2-4 semanas mostrando pantallas primefaces reales desplegadas en WAS para recibir feedback temprano y ajustar la interfaz.

2. ¿Qué principio ágil permite incorporar nuevas funcionalidades en los requisitos y cómo se aplicaría? (1 Punto)

Principio Ágil: Aceptar cambios en los requisitos

Aplicación:

La funcionalidad de descuentos rápidos se habría incorporado fácilmente, ajustando entidades SPA y servicios ESR en un sprint, sin rehacer todo el diseño inicial.

3. ¿Cómo usaría un principio ágil para reducir el impacto de la documentación extensa en el desarrollo del proyecto? (1 Punto)

Principio Ágil: Software funcional

Aplicación: